

de Mahieu Benoit
Goossens Jonathan
Groupe 12
2TL2

Développement informatique avancé :

Application

Rapport du projet

Super Quizz

<https://github.com/JonGoo/QuizzJava>

Cahier de charges (Technical Version)

Cette application est composée de 2 pages principales :

- Une page d'introduction, où le joueur peut se connecter.
- Une page de sujets et questions où le joueur peut choisir le sujet sur lequel il sera interrogé. Il peut voir ses points et son niveau dans chaque catégorie et il peut, s'il le désire, proposer une question qui pourra être ajoutée dans une des catégories.

Page d'introduction

Cette page est composée de plusieurs champs. Deux JTextArea pour le nom du jeu et les explications. Deux JTextField qui permettent d'écrire un pseudo unique et un prénom. Deux JButton, un pour créer un compte et un autre pour se connecter si le compte existe déjà.

Lorsque le joueur clique sur le bouton pour s'enregistrer, le programme va se connecter à la base de données et vérifier si le pseudo n'existe pas déjà. Si le pseudo proposé n'est pas dans la BDD, il est ajouté à celle-ci et envoie le joueur à la page de choix de sujet. Si le pseudo est dans la BDD, le programme va signaler que ce pseudo existe déjà, le joueur peut alors proposer un autre pseudo et garder son prénom.

Lorsque le joueur clique sur le bouton pour se connecter, le programme va se connecter à la BDD et vérifier si le pseudo et le prénom sont bien présent dans la BDD et s'ils se combinent. S'ils ne sont pas bons, le programme affiche que le pseudo ou le prénom n'est pas bon. S'ils sont juste, le programme envoie le joueur à la page de choix de sujet.

Page des sujets + questions

Partie sujets

Cette page est composée de plusieurs champs, un avec le pseudo, les points et les niveaux du joueur, un autre avec un bouton pour proposer une question à ajouter dans la BDD et un champ de choix de sujet sur lequel le joueur sera interrogé.

Lorsque le joueur clique sur le bouton pour proposer une question, un formulaire composé de 5 JTextField s'affiche et le joueur peut soumettre sa question en indiquant la bonne réponse en premier et les mauvaises par la suite. Il y a deux boutons en dessous de ce formulaire, un pour valider et envoyer la question dans la BDD et un autre pour retourner au menu de choix de sujet !

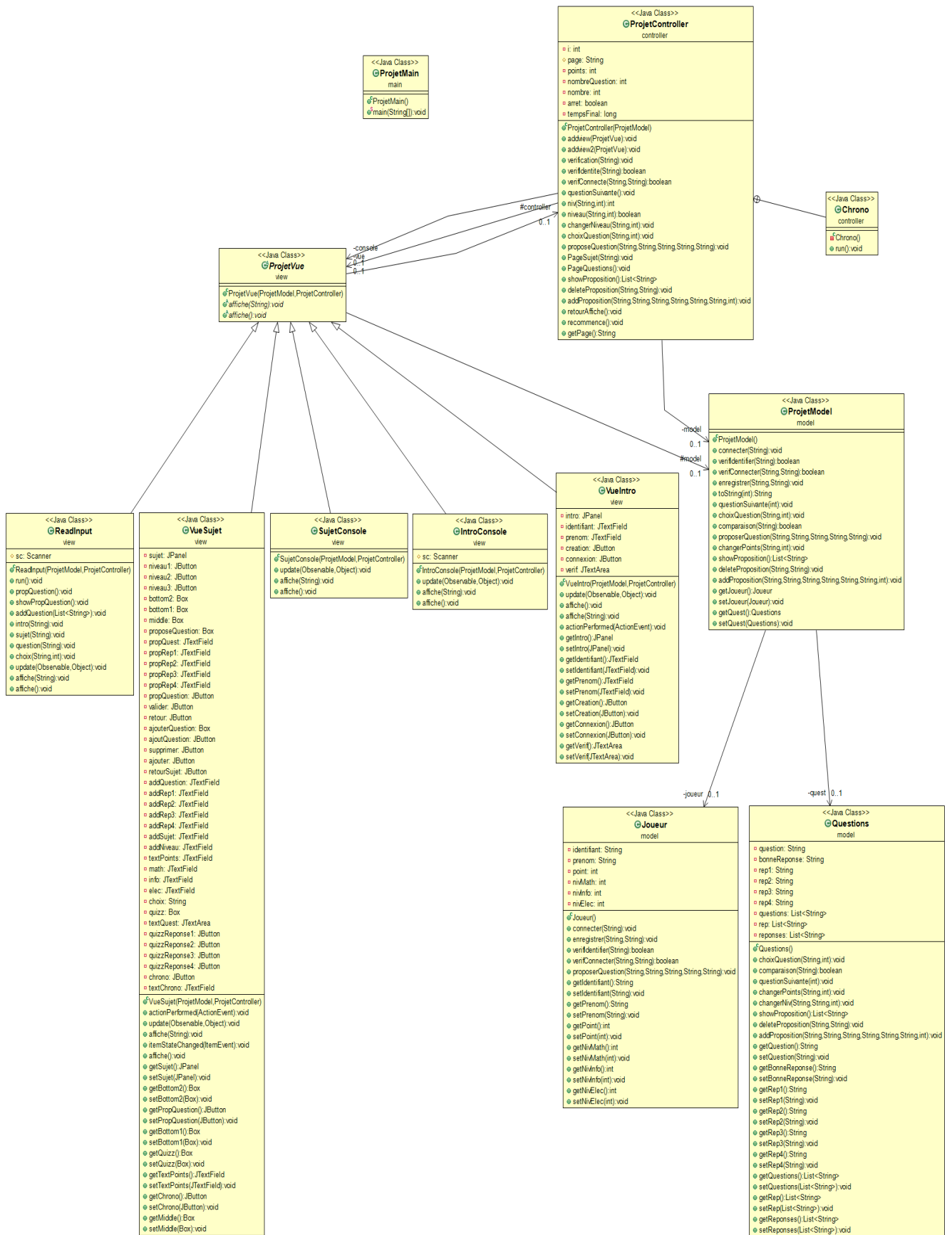
La partie sujet est composée d'une liste déroulante. Le joueur va choisir entre les différents sujets proposés. Trois boutons apparaissent pour choisir le niveau des questions. Si le joueur veut débloquent le niveau 2, il devra dépenser 200 points et 400 points pour le niveau 3 ! Une fois qu'il a cliqué sur l'un de ces boutons, La partie question apparait directement.

Partie questions

Cette partie est constitué d'un champ JTextArea pour la question et de quatre JButton pour les réponses. Il y a également un JButton pour le compte à rebours (sous cette forme pour l'esthétique).

Les questions sont choisies aléatoirement dans la BDD en fonction du sujet et du niveau que le joueur a sélectionné précédemment. Les réponses sont quant à elles aussi mélangées. Le programme va poser 5 questions et changera les points du joueur s'il a bien répondu. Le compte à rebours est initié à vingt secondes. Le joueur aura ce temps-là pour répondre à la question sinon il ne gagnera pas de points. Une fois que les questions sont toutes passées, le joueur sera renvoyé à la page de choix de sujet, il pourra donc continuer à jouer !

Diagramme UML :



Implémentations :

Nous avons décidé d'utiliser une base de données pour enregistrer les Joueurs et nos questions. Par après, il nous est venu l'idée de laisser la possibilité aux joueurs de proposer une question à ajouter à la liste. On a dû apprendre JDBC et on a remarqué que c'était très simple à implémenter.

Les consignes nous obligeaient d'implémenter le pattern MVC... Comme nous avons déjà commencé le projet, l'intégration du pattern fut difficile mais une fois finie, cette implémentation est pratique mais peut être trop verbeuse.

Enfin, nous avons utilisé Swing pour l'interface graphique car on l'a vu en cours et qu'on ne voulait apprendre une autre partie... Ce n'était pas la plus pratique mais nous y sommes arrivés.

Problèmes rencontrés :

- L'interface graphique Swing n'est pas optimale dans la façon de structurer les éléments qui la compose (Les JTextField, JTextArea, ...)
- Notre projet utilise plusieurs JFrame, pour passer de l'une à l'autre nous avons eu pas mal de soucis pour l'associer avec le pattern MVC !

Améliorations éventuelles :

- Implémenter une partie DUEL/BATTLE où 2 joueurs peuvent miser un certain nombre de points et être interrogé sur un sujet. Celui qui répond le mieux aux questions l'emporte !
- Mettre la base de données sur un serveur pour que n'importe qui puisse jouer au jeu, depuis n'importe quel ordinateur.
- Lors de la proposition de questions par un joueur, la question sera associée au joueur et si cette question est conservée pour être utilisée, ce joueur gagner un certain nombre de point.

Conclusions personnelles :

- **Benoit de Mahieu :**

Ce projet m'a permis de bien comprendre l'orienter objet que Java propose. J'avais du mal au début à voir l'intérêt de ce système mais je remarque la force lorsque j'avais plus précisément dans le projet.

Mon frère m'a montré la connexion BDD et j'ai compris rapidement comment l'utiliser. Grâce à cela nous avons avancé dans le projet sans que ce soit un obstacle. Notre BDD est un pilier de l'application avec le pattern MVC.

Il m'a fallu beaucoup de temps pour gérer le pattern MVC parce que les deux Jframes que nous utilisons a causé beaucoup de soucis.

Après cela j'ai regardé quelques vidéos sur swing pour continuer mais cela reste encore complexe à gérer.

J'ai bien compris les Thread après le TP consacré là-dessus et ce fut la dernière difficulté que j'ai rencontrée.

- **Jonathan Goossens :**

Ce projet m'a permis d'entrevoir ce qu'est le travail d'équipe dans le développement d'application. Ce n'est pas évident de se répartir le travail, surtout pour moi qui suis très discret !

Java est un langage intéressant qui ouvre énormément de porte, surtout grâce à l'orienté objet. Par contre, j'ai trouvé que SWING n'était pas la manière la plus aisée de faire une interface graphique... Par la suite, je me pencherai un peu plus sur JavaFX pour voir si c'est plus simple.

Enfin, ce projet m'a permis de revoir les bases de données et de les voir d'une autre manière qu'une page WEB. C'est génial de voir que notre BDD et notre programme sont liés et que l'on peut tout faire grâce à ça !